γ – LASER Mode
Beam Monitoring & Scintillation Detectors

John Hallford

09/07/2020

LUXE

DESY.

For the simplest Bremsstrahlung interaction, this is necessarily:

$$E_\gamma = E_{beam} - E_e$$

For two successive Bremsstrahlung interactions:

$$E_e = E_{beam} - E_{\gamma 1} - E_{\gamma 2}$$

We take the ‘first-order’ Brem interaction dominates and we use it to calculate E-gamma

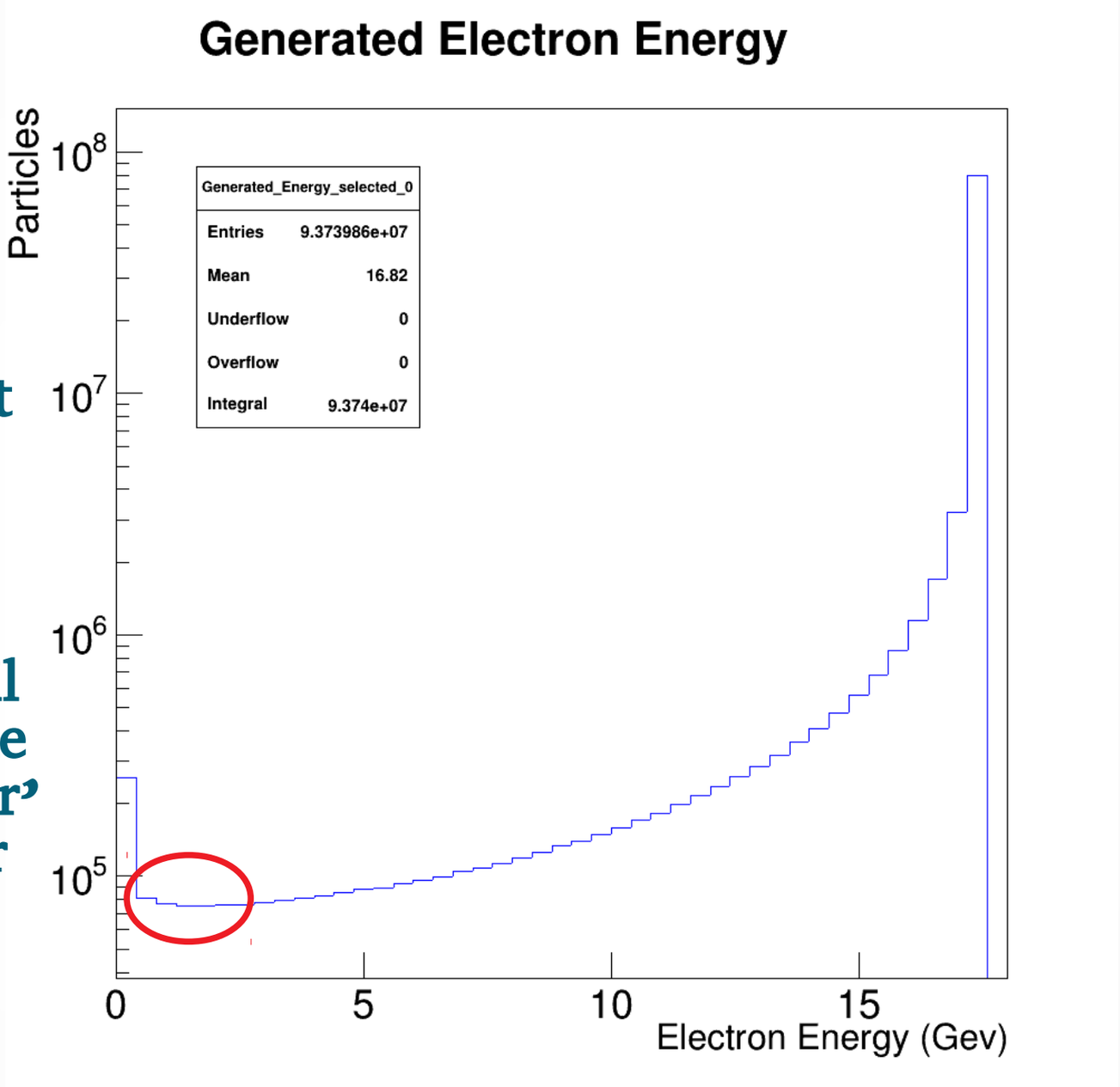
Any emitted gamma can undergo pair production, where as a statistical average the energy of each positron is one half of the gamma energy

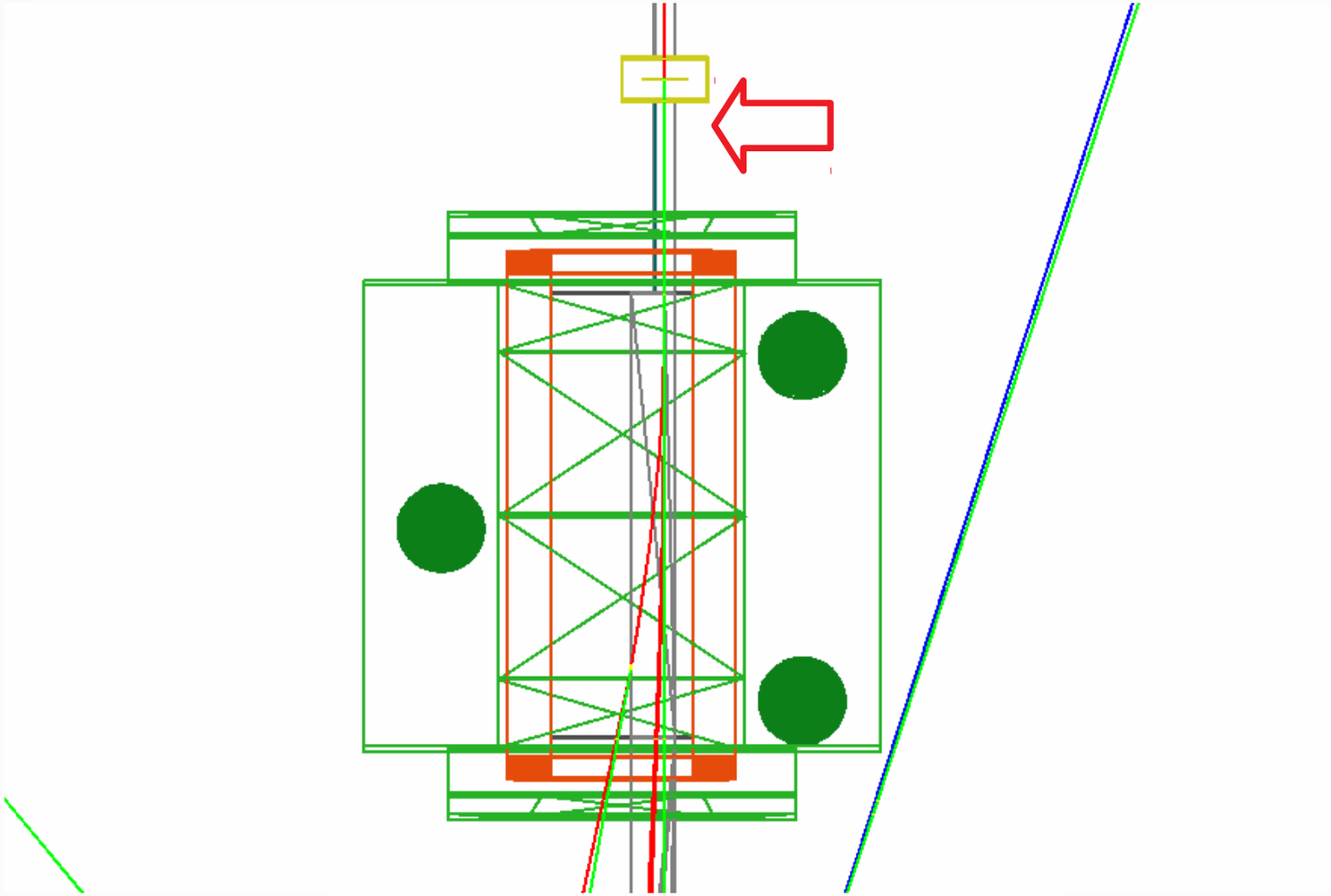
This trend is of course weaker in correlation to gamma distribution and the creation of positrons far less frequent than Bremsstrahlung

Previously…

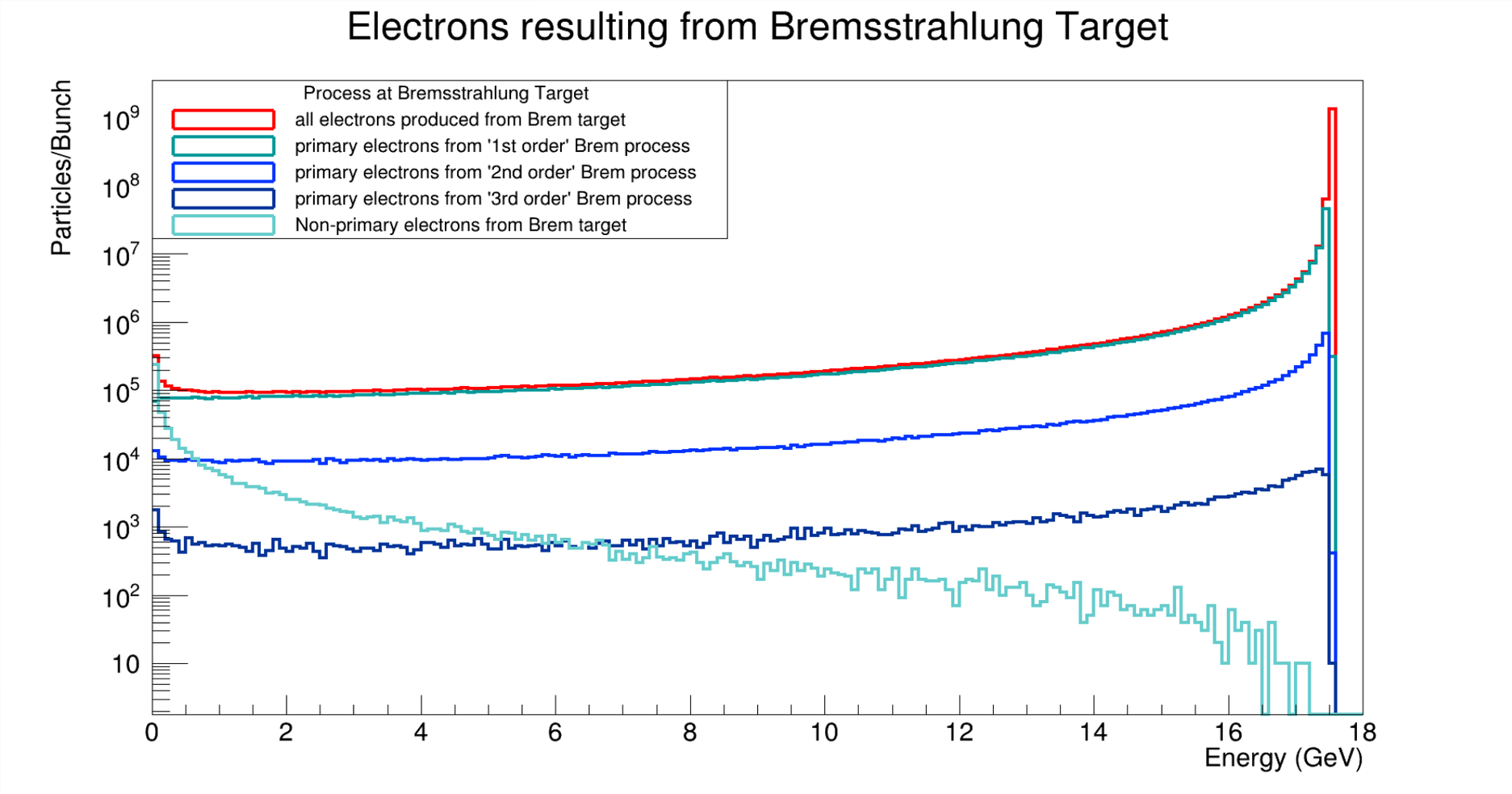
Our Detector acceptance/magnet aperture limits the region of electron energy distribution – roughly 1.6 GeV to 15 GeV

There is a kink in the low-E tail of the e- energy spectrum on the right. Is this due to ‘higher-order’ Bremsstrahlung interactions? Or Something else?





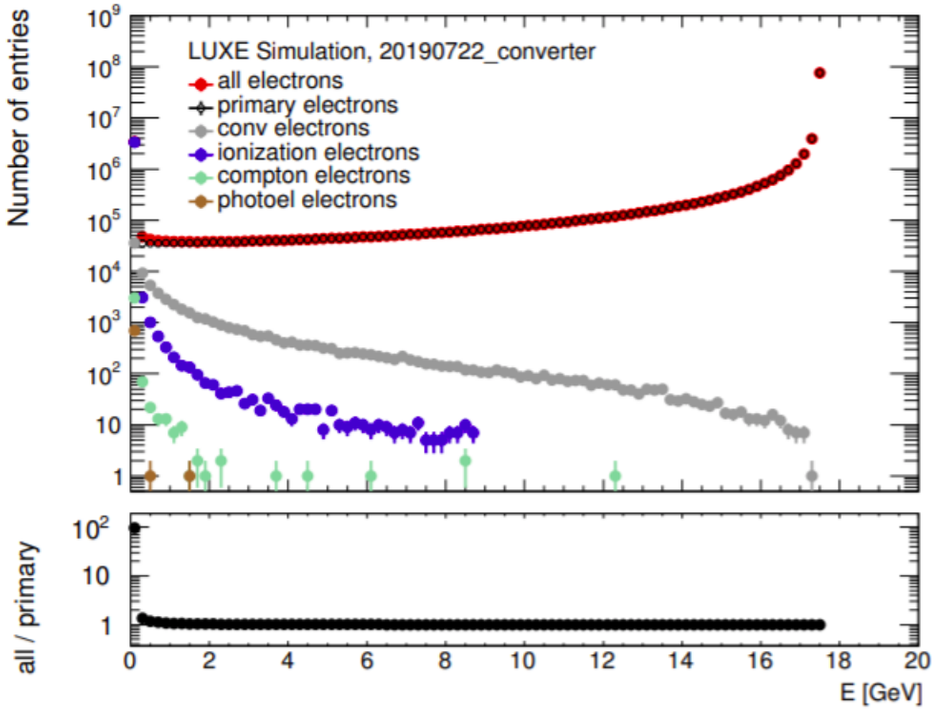
So.. we amend lxsim.cc to measure the output of the Brem target, i.e. count the particles entering the immediate next downstream volume, and sort event – by event into ‘order’ of Brem interaction



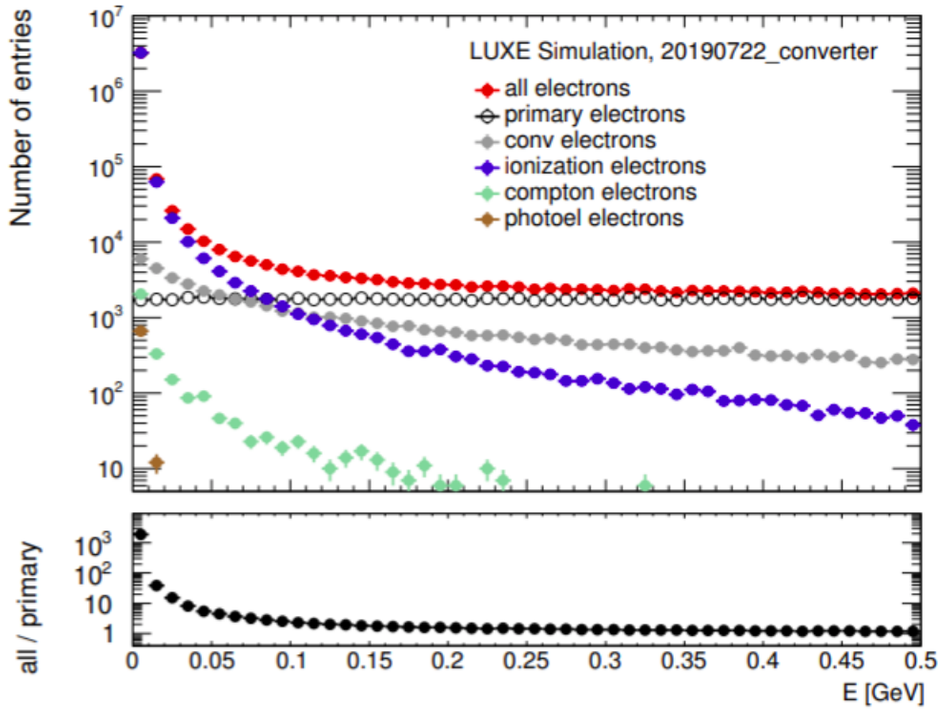
The bump in electron energy is well explained by non-primary electrons, and not a shift in energy for 'Higher-order' Brem Processes.

My idea that the 'higher-order' processes result in lower energy e- is only faintly shown.

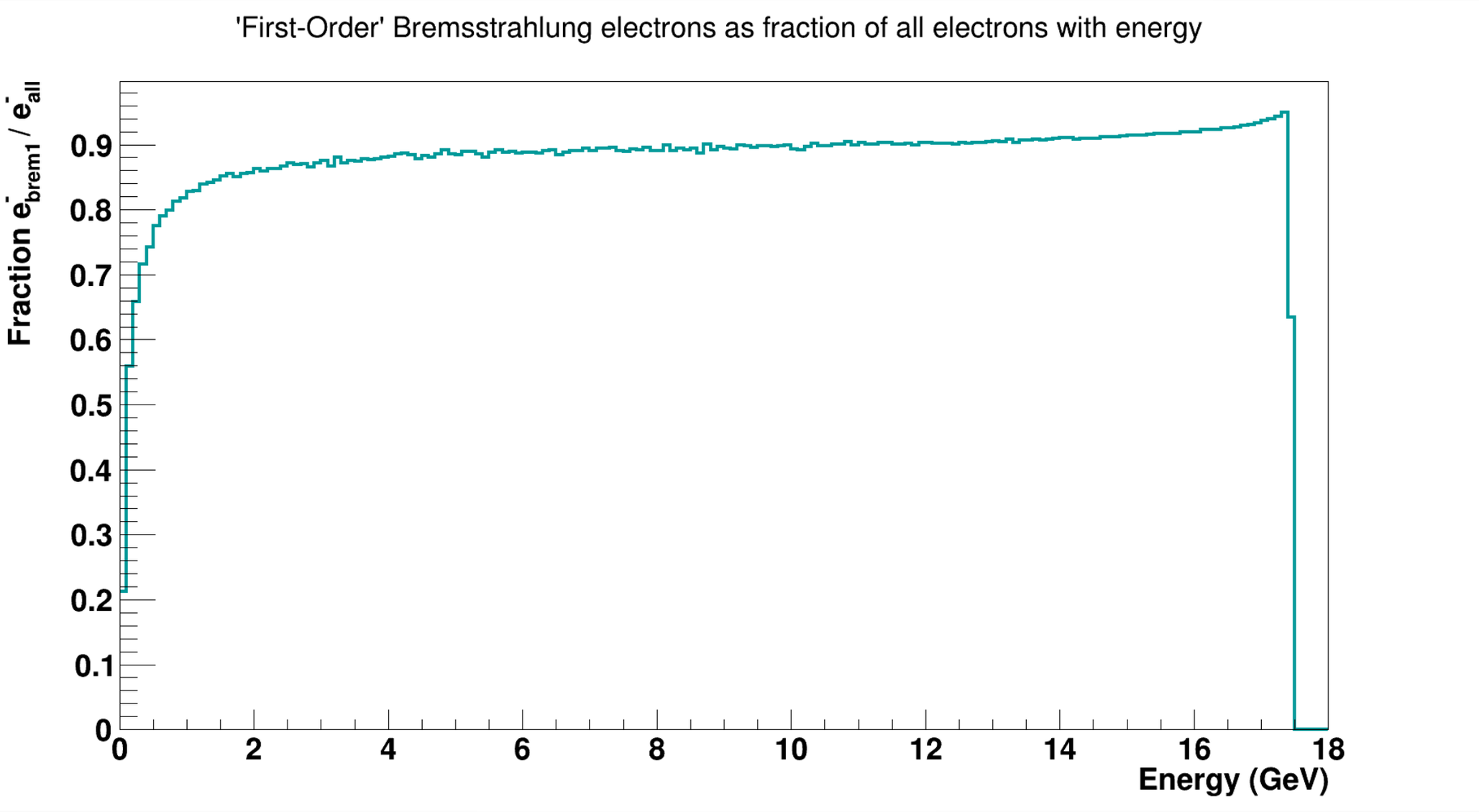
complete spectrum [0-20 GeV]



low energy spectrum [0-0.5 GeV]

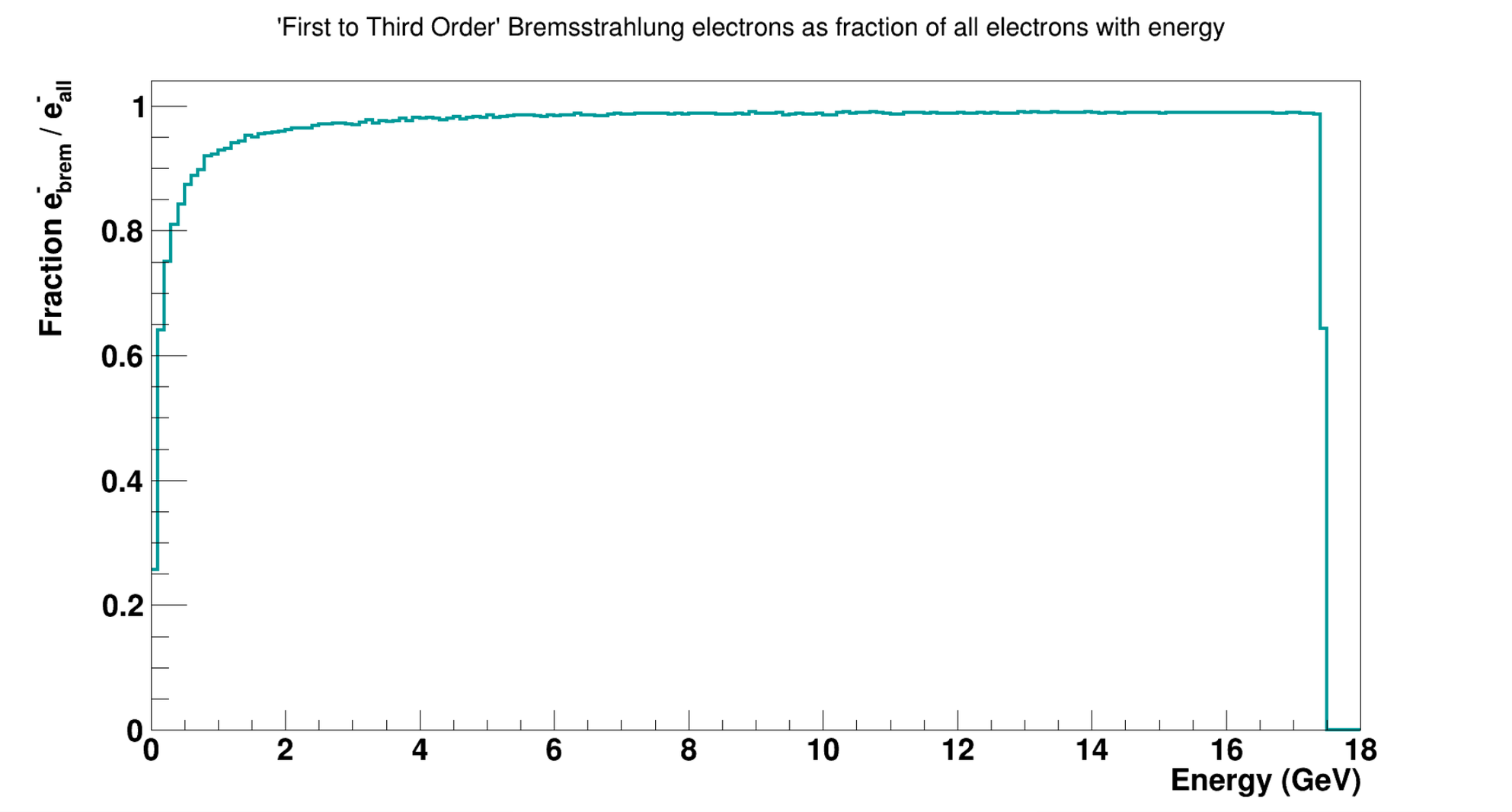


- electrons dominated by ionization for $E < 200$ MeV, primaries for $E > 200$ MeV
- conversions/primaries $< 3\%$ for E above 2 GeV
- compton/photo-electrical electrons sub-dominant everywhere



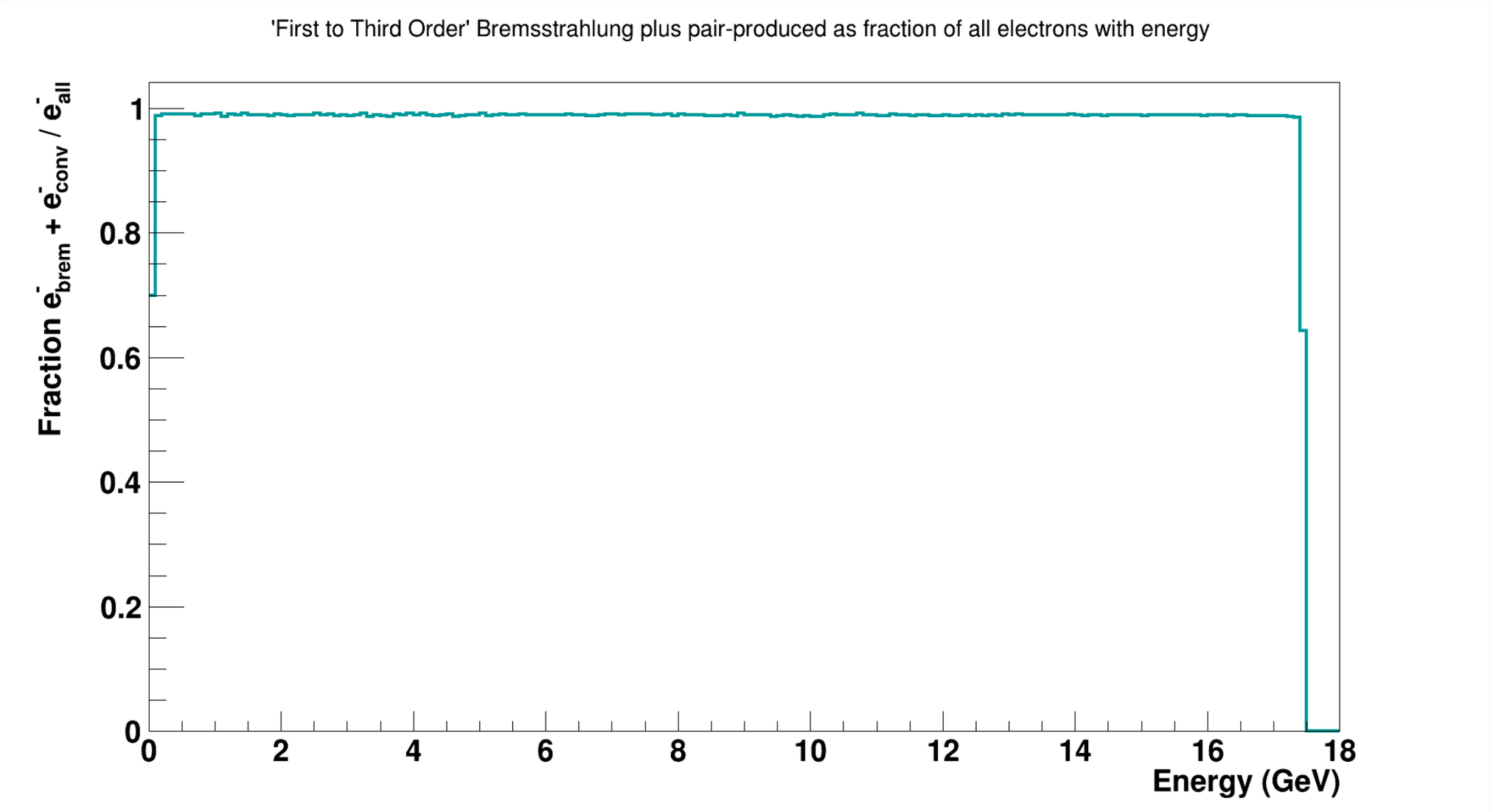
At 1.6 GeV ~ 85%

at 15 GeV ~ 91%



At 1.6 GeV ~ 95%

at 15 GeV ~ 99%



At 1.6 GeV ~ 99%

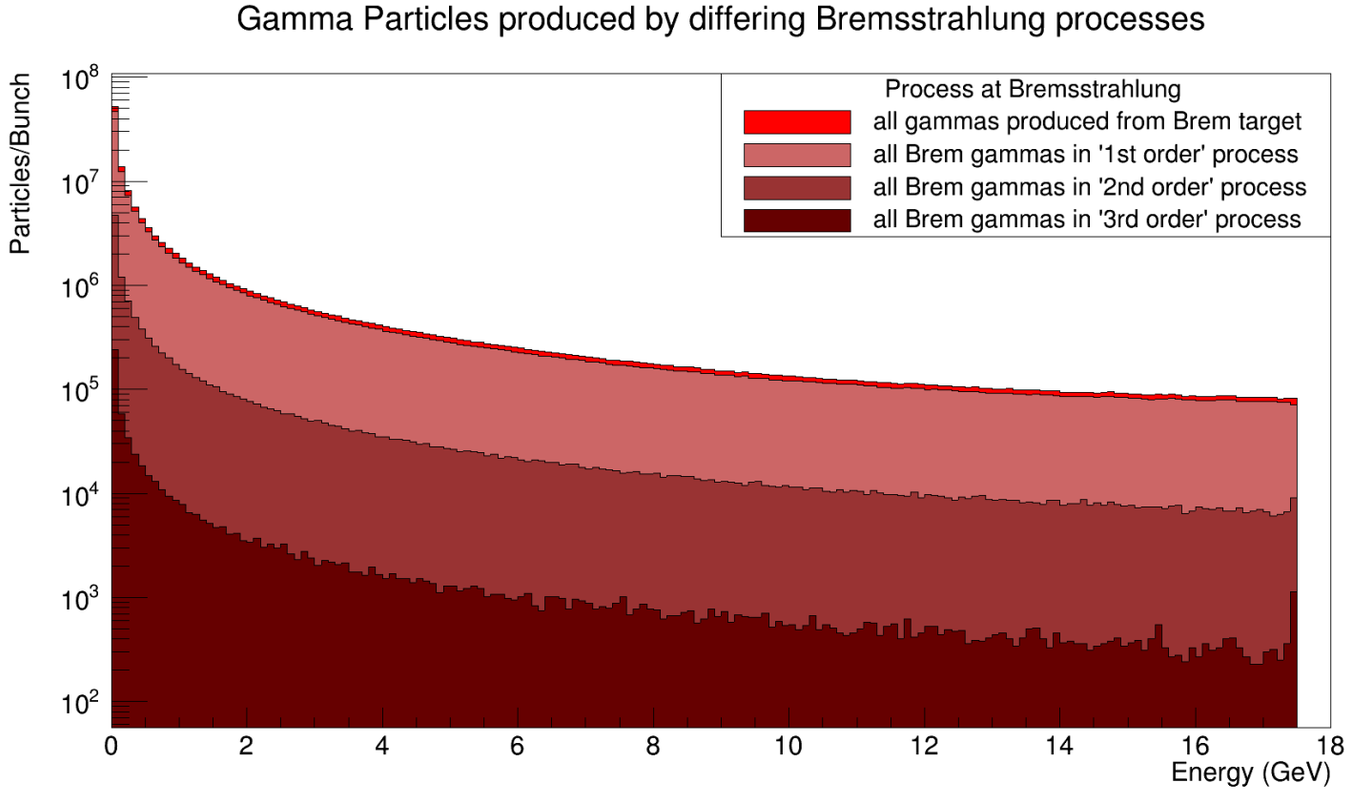
at 15 GeV ~ 99%

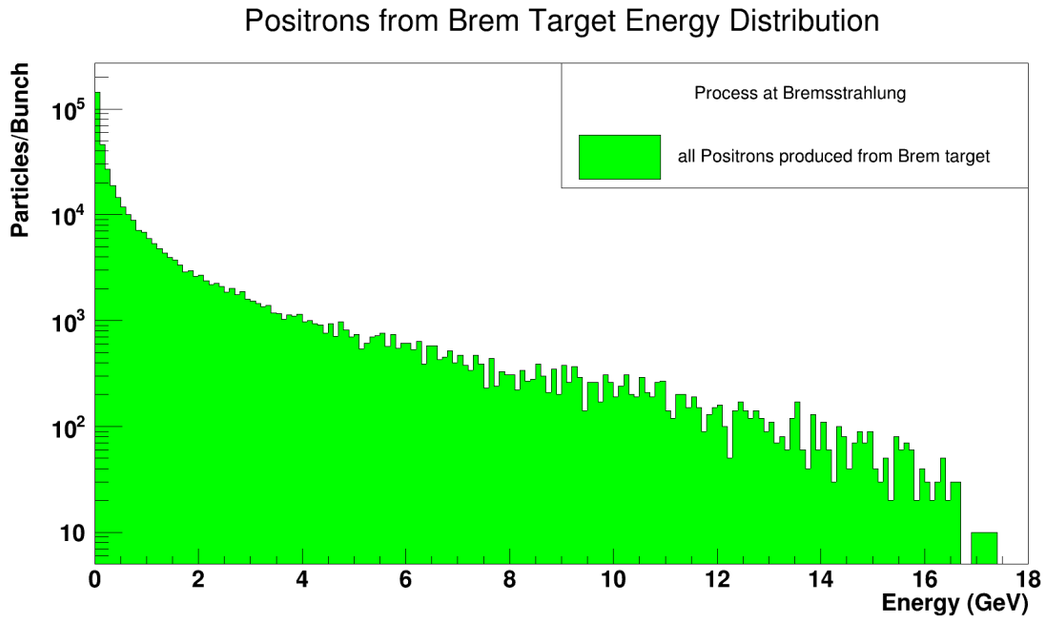
We have distributions of interest for gamma, positrons

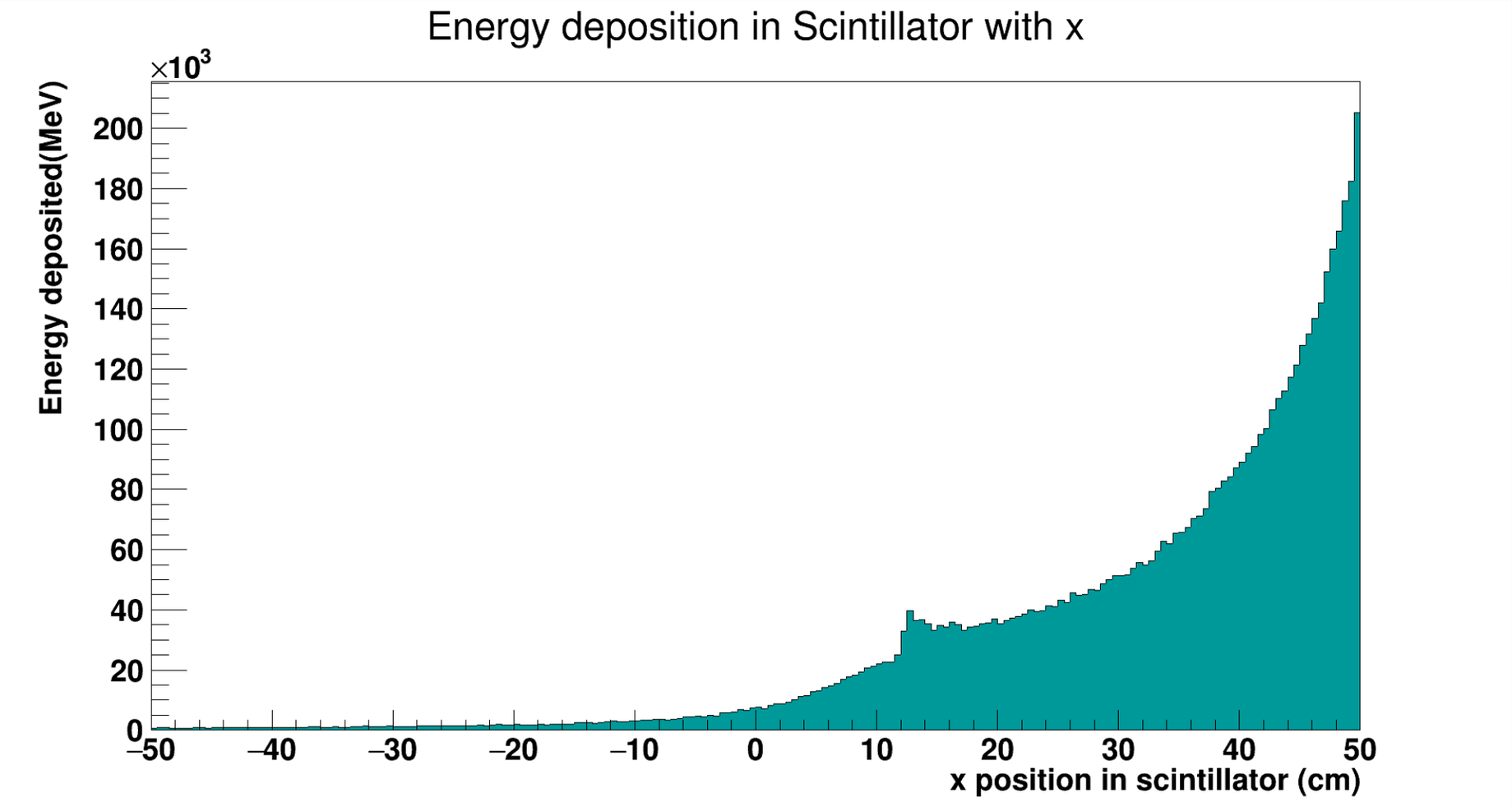
Should use e-, e+ to reconstruct gamma distribution and see how well they compare

These were completed on a new install of GEANT4 on the UCL HEP cluster! Can run jobs there.

Used a modified lxsim which removes shielding and everything down-stream of Brem target/detectors. This increases simulation speed by ~100 times, but is not useful for anything but Brem target detector studies. We can simulate 10^6 events in ~50 mins/thread, so can simulate a full bunch of 1.5×10^9 primaries over 150 threads in 5 hours.







Also – we see a bump in the energy deposition at what I believe to be the limit of the aperture, again. I will investigate more and run a simulation with the screen closer to the magnet intending to minimise this

lxsim Random seed note:

A small change in lxbeamsim.cc –
Can add two arguments which the program will take as two random engine seeds.
The first argument still defines thread count. Next two optional arguments can change RNG seeds.
Accessible therefore from bash scripts

```
john@john-Laptop:~/lxsim-build$ ./lxbeamsim run.mac 4 8000 9000

*************************************************************
 Geant4 version Name: geant4-10-06-patch-01 [MT]   (14-February-2020)
  << in Multi-threaded mode >>
                       Copyright : Geant4 Collaboration
                      References : NIM A 506 (2003), 250-303
                                 : IEEE-TNS 53 (2006), 270-278
                                 : NIM A 835 (2016), 186-225
                             WWW : http://geant4.org/
*************************************************************

===== lxbeamsim is started with 4 threads =====
===== lxbeamsim seeds are : 8000 and 9000 =====
```

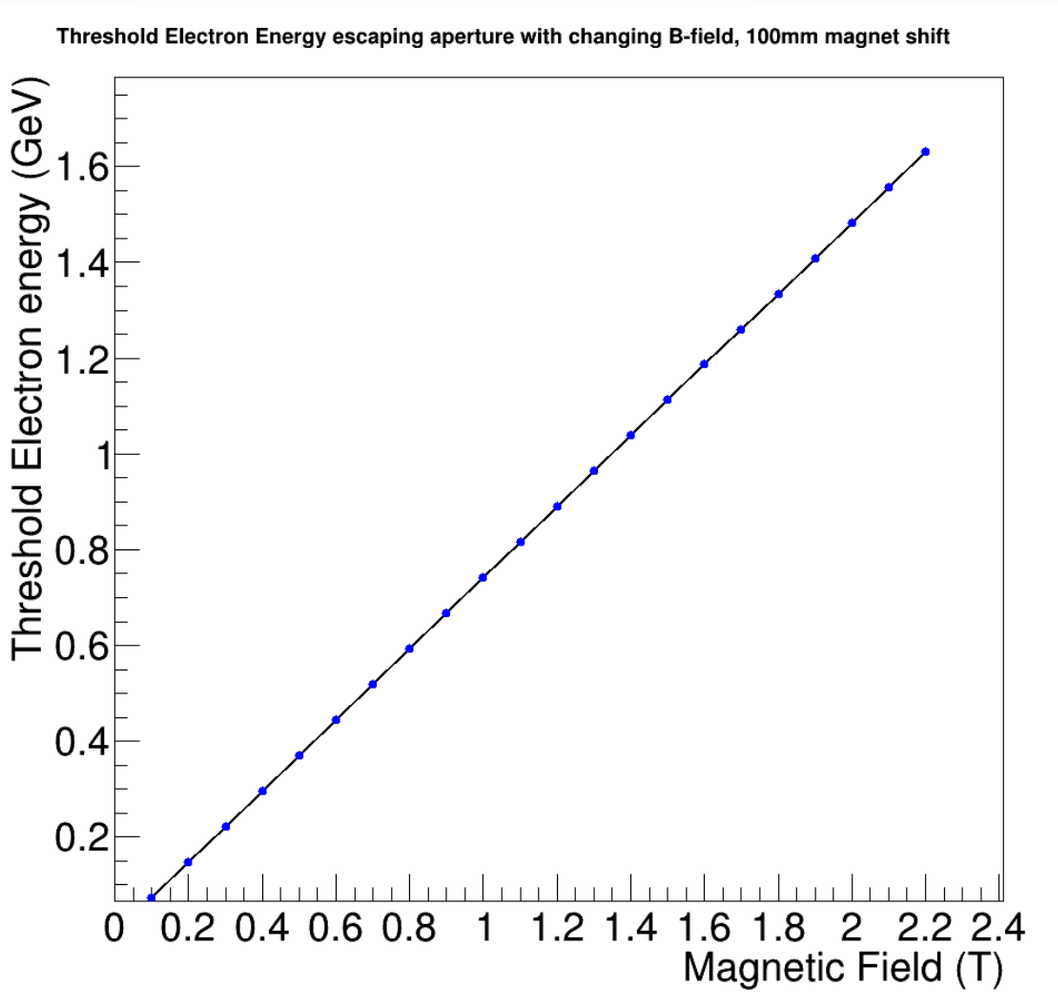
Example lines for script using linux RNG:

```
random1=$(shuf -i 0-99999 -n 1)

random2=$(shuf -i 0-99999 -n 1)

./lxbeamsim run.mac $random1 $random2
```

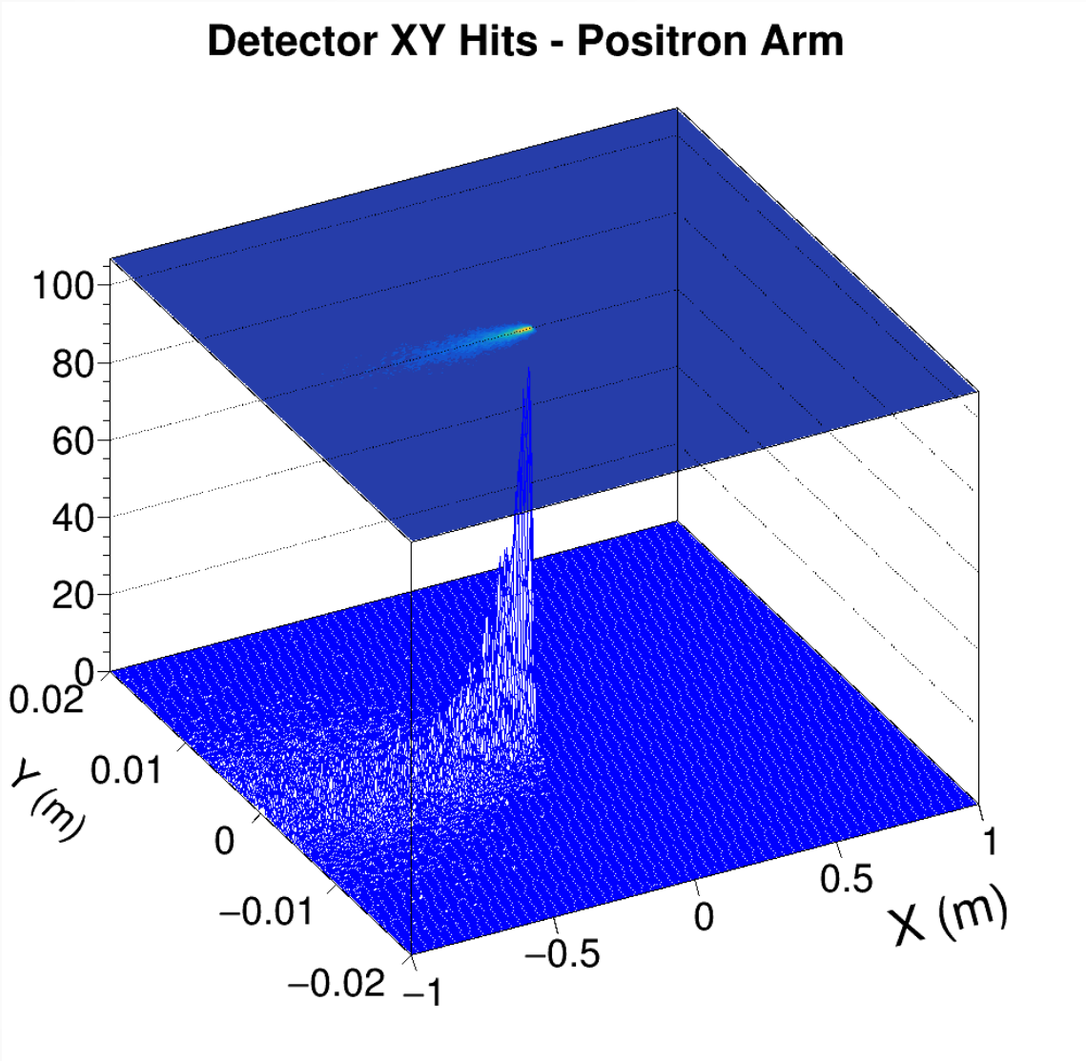
backup

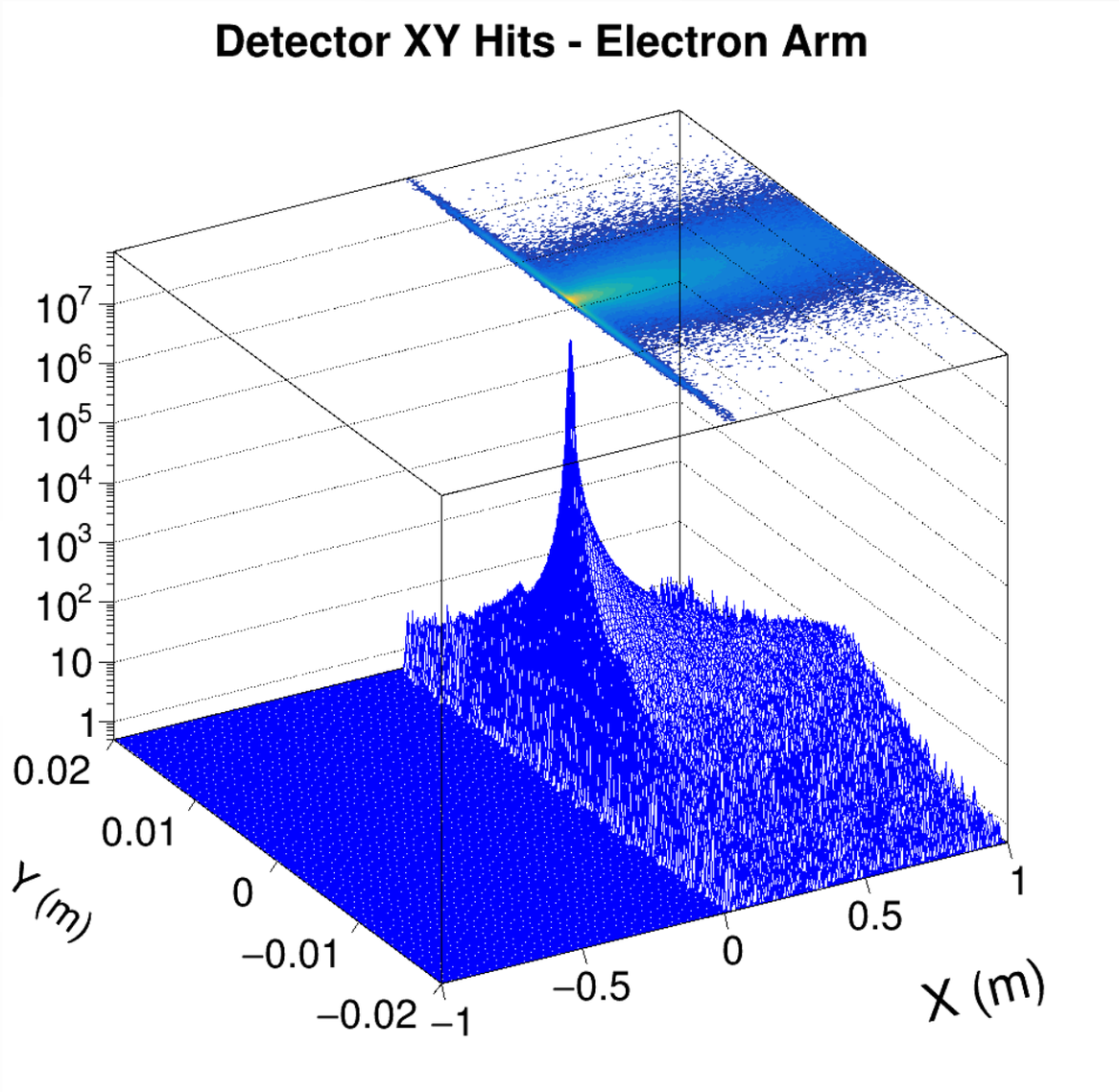
lxsim Random seed note:

```
int main(int argc,char** argv) {

  long seeds[2] = {12345, 66666};

  if (argc == 4 || argc == 5){

    long random [2] = {G4UIcommand::ConvertToInt(argv[argc-2]), G4UIcommand::ConvertToInt(argv[argc-1])};
// C++ is not happy if we try to use these arguments in a non-initialised vector
    seeds[0] = random[0]; seeds[1] = random[1];
  }

  //choose the Random engine

  G4Random::setTheEngine(new CLHEP::RanecuEngine);
  G4Random::setTheSeeds(seeds, 3);

  // Construct the default run manager
#ifdef G4MULTITHREADED
  G4MTRunManager* runManager = new G4MTRunManager;
  //G4int nThreads = G4Threading::G4GetNumberOfCores();
  G4int nThreads = 1;
  if (argc==3 || argc==5) nThreads = G4UIcommand::ConvertToInt(argv[2]);
  runManager->SetNumberOfThreads(nThreads);
  G4cout << "===== lxbeamsim is started with "
         <<  runManager->GetNumberOfThreads() << " threads =====" << G4endl;

#else
  G4VSteppingVerbose::SetInstance(new SteppingVerbose);
  G4RunManager* runManager = new G4RunManager;
#endif
    G4cout << "===== lxbeamsim seeds are : " << seeds[0] << " and " << seeds[1] << " =====" << G4endl;
```



B-field	Magnet off-axis shift	Threshold energy
2.2 T	0 mm	~2.2 GeV
2.2 T	100 mm	~1.6 GeV


Detector XY Hits - Positron Arm
100
80
60
40
20
0
0.02
0.01
0
−0.01
−0.02
Y (m)
−1
−0.5
0
0.5
1
X (m)


Detector XY Hits - Electron Arm
10^7
10^6
10^5
10^4
10^3
10^2
10
1
0.02
0.01
0
−0.01
−0.02
Y (m)
−1
−0.5
0
0.5
1
X (m)